

WebSphere Enterprise Service Bus and WebSphere Integration Developer



Rob Phippen

IBM Senior Technical Staff Member

Chief Architect, WebSphere Enterprise Service Bus



THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Agenda

- Introduction
 - Existing business problems
 - SOA and ESB concepts
- WebSphere Enterprise Service Bus
 - A Closer look at using WebSphere ESB and WID
 - Runtime management
 - Latest updates
 - Roadmap for the future
- Summary



- Multiple languages
- Multiple protocols
- Heterogeneous environments
- Tight coupling
- Inflexibility of existing systems

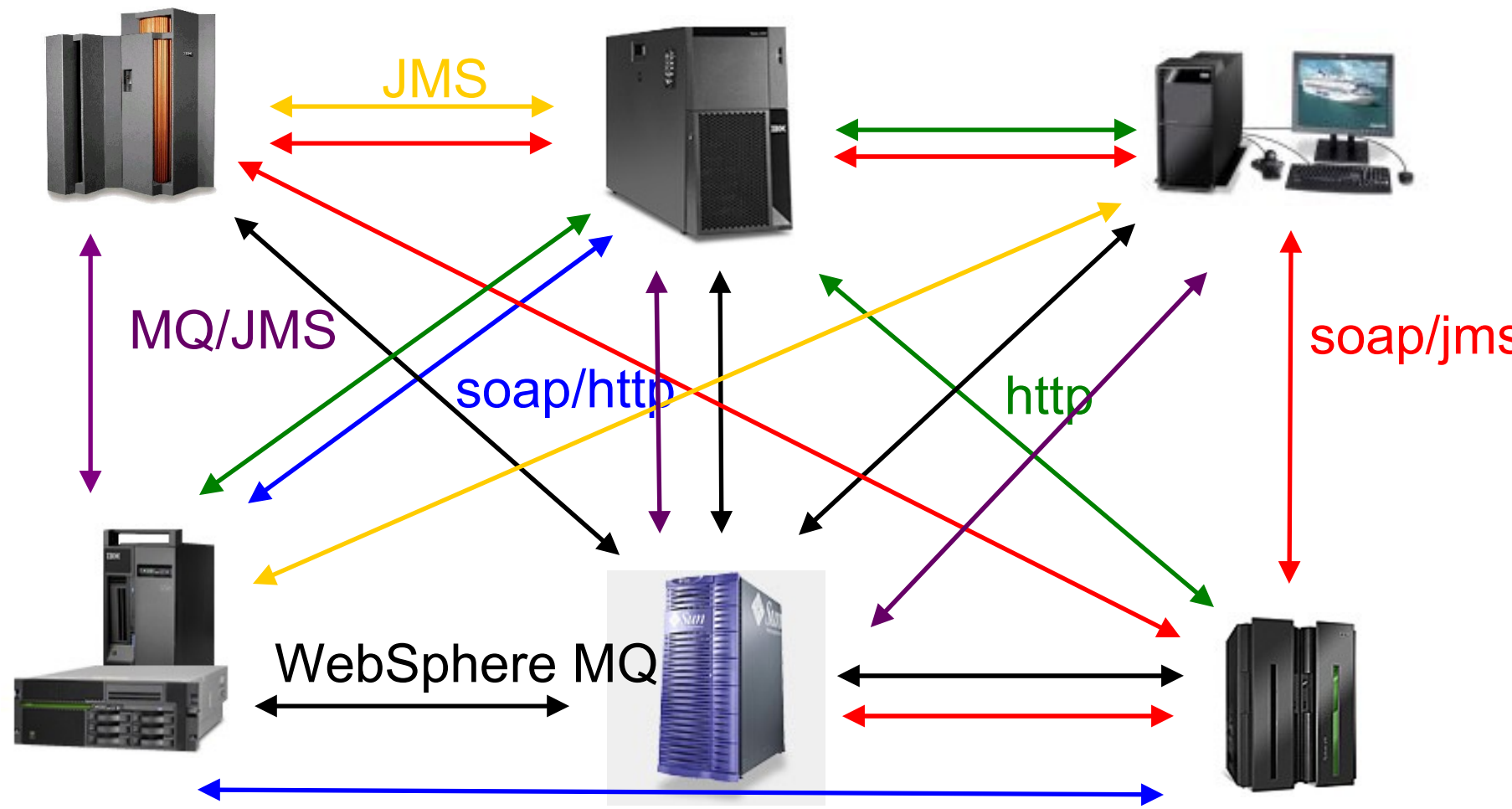
```
public class Warehouse {  
  
    public int getStock (String partNumber) {  
        database DB = new database();  
        return DB.dblookup(partNumber);  
    }  
}
```

```
01 WAREHOUSE-QUOTE-STRUCTURE.  
05 UTF-HEADER.  
    10 CICS-RESP-MAJOR-CODE      PIC X(00004).  
    10 CICS-RESP-MINOR-CODE     PIC X(00004).  
    10 CICS-RESP-MESSAGE       PIC X(00072).  
    10 HEADER-FILLER           PIC X(00020).  
05 UTF-APPLICATION-DATA.  
    10 INQUIRY-NUMBER          PIC X(00010).  
    10 PART-NUMBER            PIC X(00010).  
    10 INVENTORY-NUMBER       PIC 9(00008).  
    10 PATH-EXECUTION-RECORD  PIC X(00300).
```

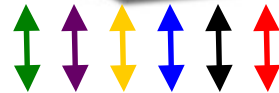
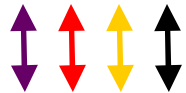
```
class Warehouse:  
    def inventory(self):  
        self.getInventory()  
  
    def status(self):  
        print 'OK'
```

```
#include <iostream>  
using namespace standard;  
  
int main () {  
    char prod[] = {'W', 'E', 'S', 'B', '/0' };  
    cout << prod;  
    return 0;  
}
```

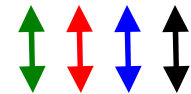
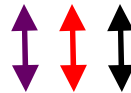
Existing Business Problems



Removing complexity



Enterprise Service Bus



The Enterprise Service Bus

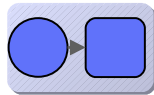
An Enterprise Service Bus (ESB) is a flexible connectivity infrastructure for integrating applications and services.

An ESB performs the following between requestor and service



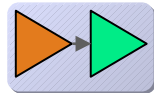
MATCHES & ROUTES

communications between services



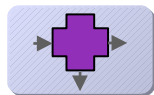
CONVERTS

between different transport protocols



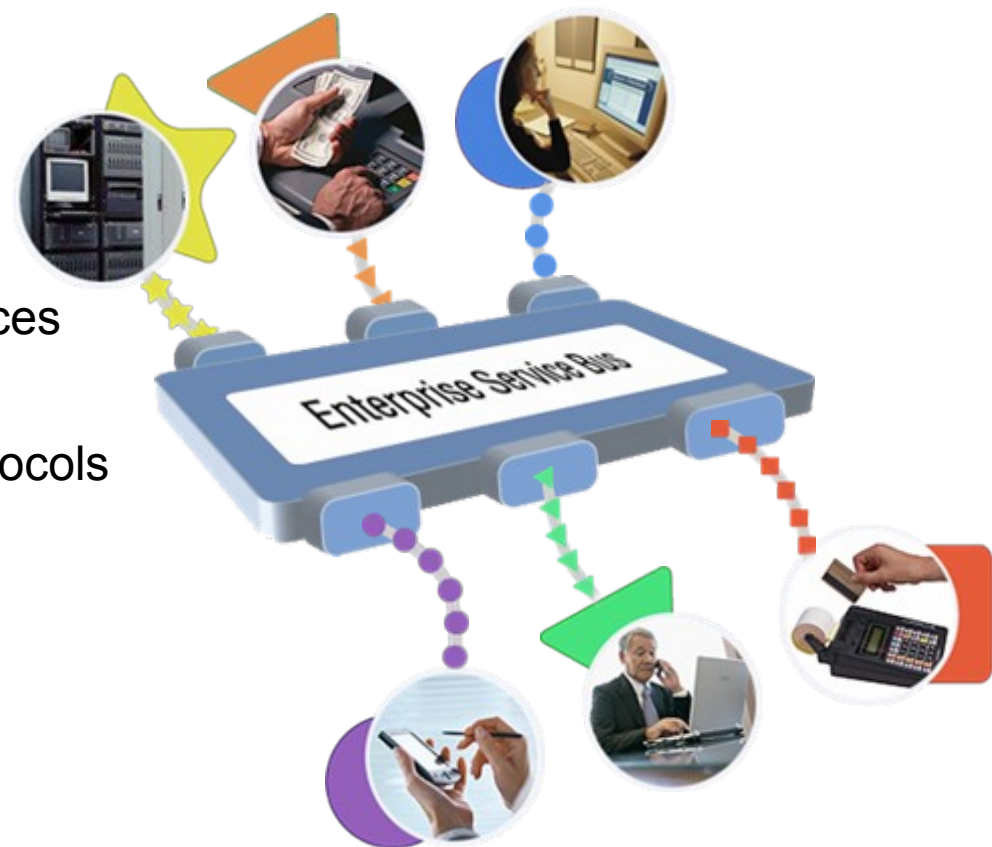
TRANSFORMS

between different data formats

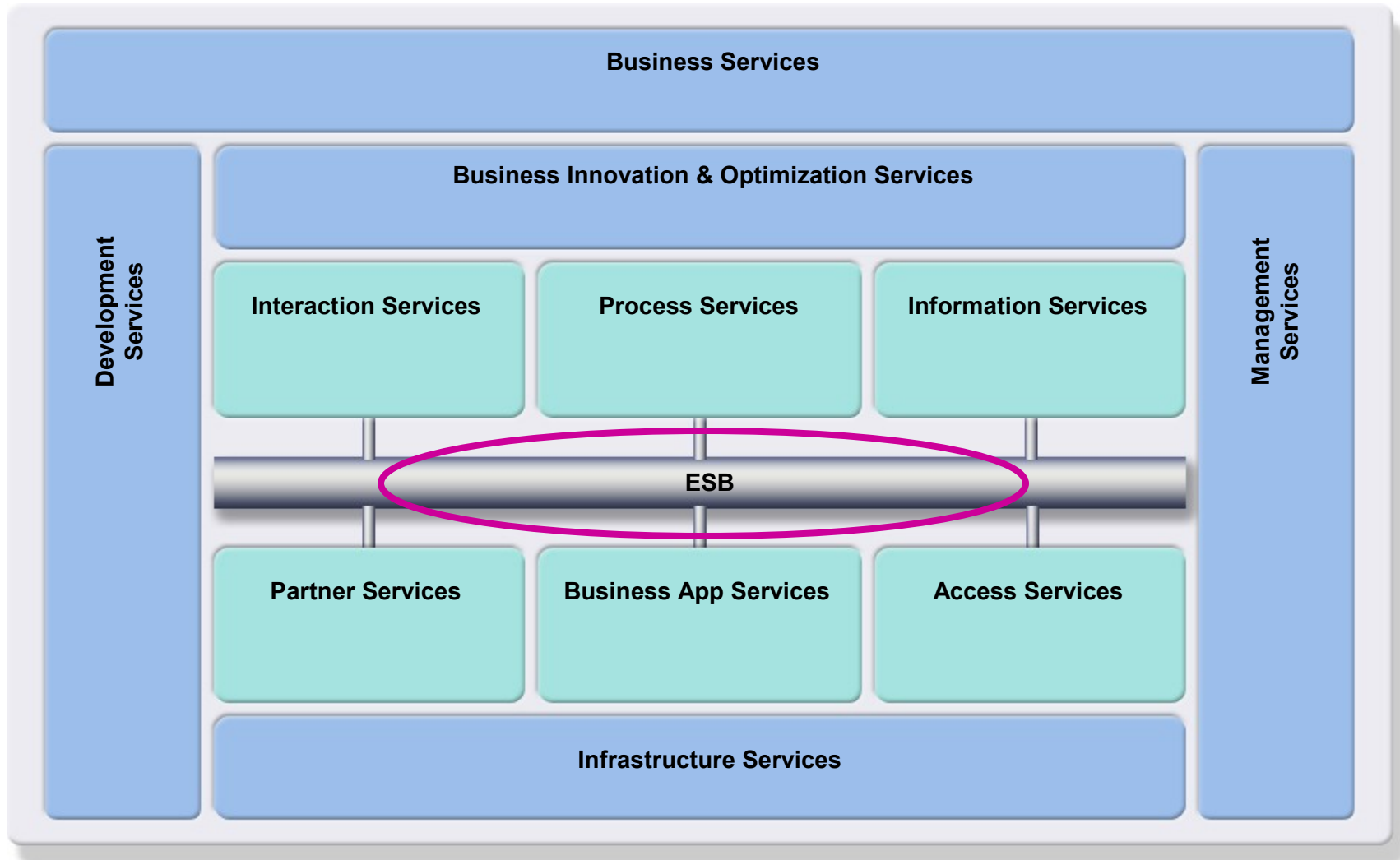


IDENTIFIES & DISTRIBUTES

business events



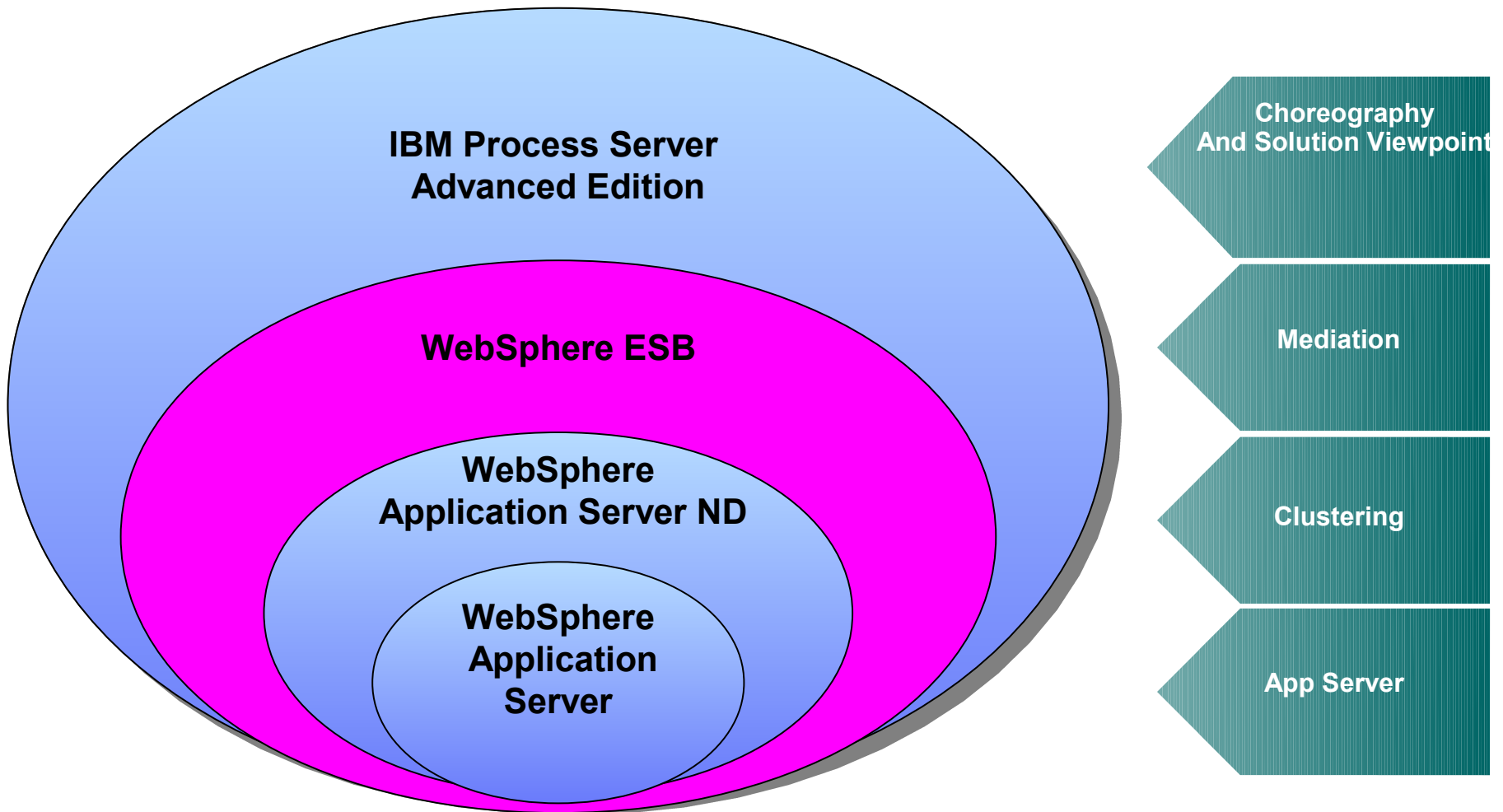
SOA and the Enterprise Service Bus



Fundamentals

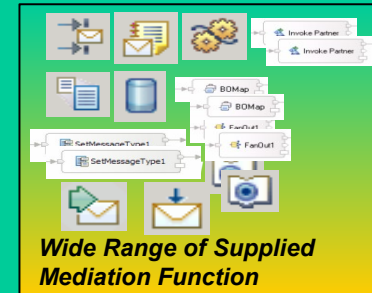
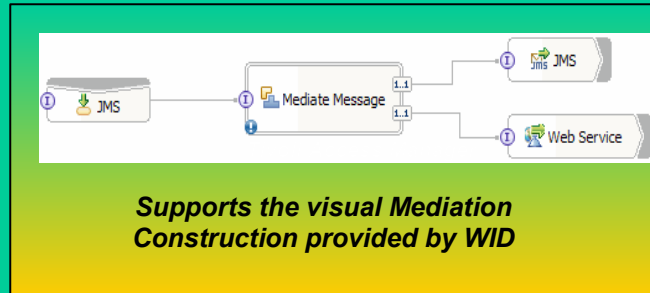


WebSphere ESB Essentials: Family Values...



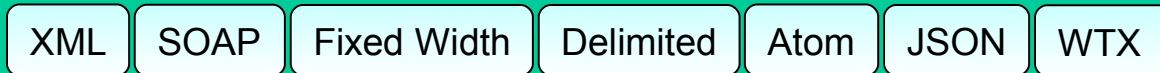
WebSphere ESB

Mediation Support
(transformation, logging, routing, etc)

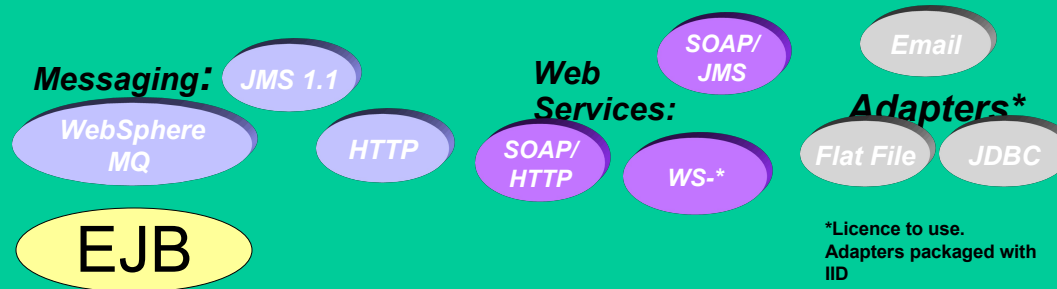


Format Support

(convert from any to any)



Protocol Support
(connect from any to any)



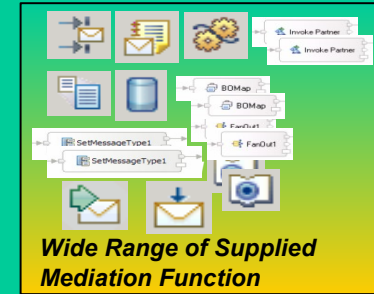
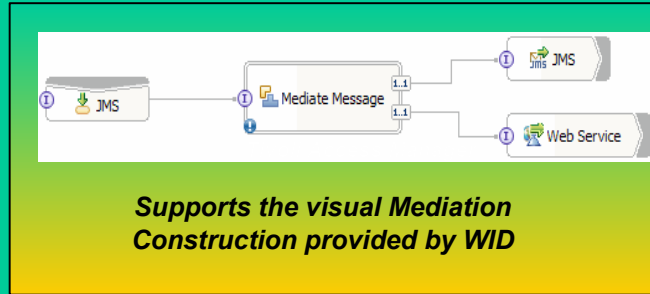
EJB

Enterprise QOS



WebSphere ESB

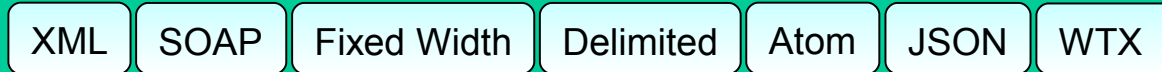
Mediation Support
(transformation, logging, routing, etc)



WebSphere Service Registry and Repository

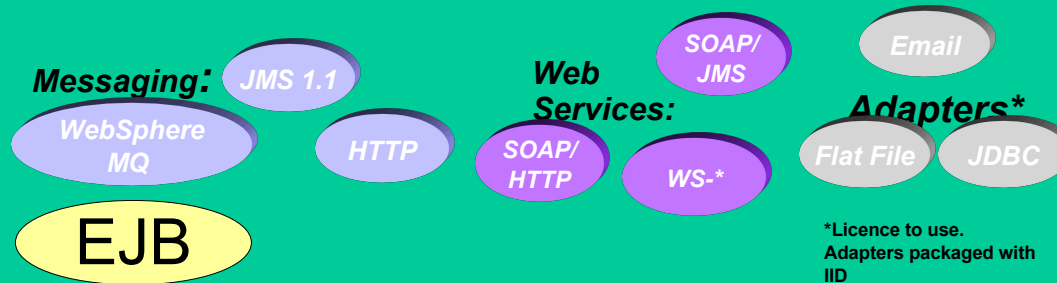
Format Support

(convert from any to any)



WebSphere Transformation eXtender

Protocol Support
(connect from any to any)



WebSphere Adapters

Enterprise QOS

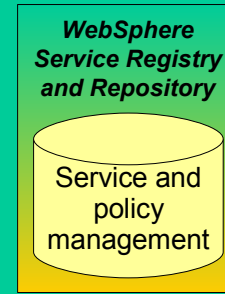
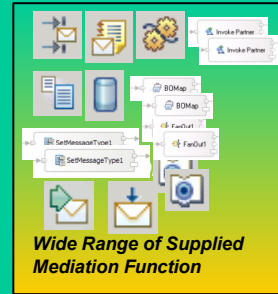
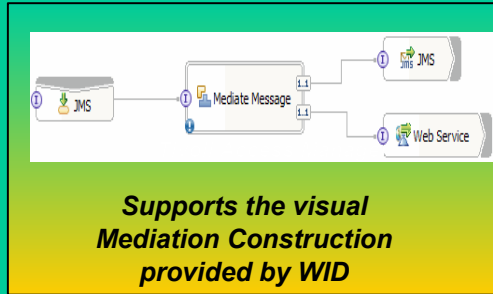




WebSphere Integration Developer

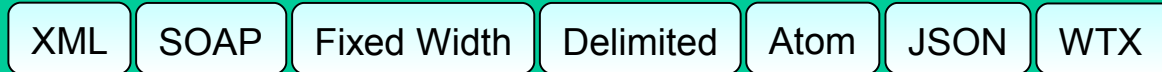
WebSphere Enterprise Service Bus Registry Edition

Mediation Support
(transformation, logging, routing, etc)



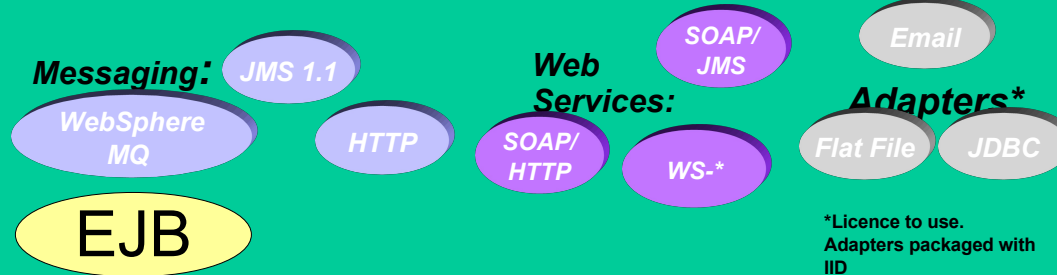
Format Support

(convert from any to any)



WebSphere Transformation eXtender

Protocol Support
(connect from any to any)



WebSphere Adapters

*Licence to use. Adapters packaged with IID

Enterprise QOS



A Closer Look



User Roles and Tasks

ESB Solution

WebSphere ESB

WebSphere Integration Developer

Solution Administrator



- Operates in “administration space”
- Understands basic interaction patterns of business processes
- Focuses on deployed mediation modules and their external interactions

Integration Developer



- Operates in “development space”
- Understands semantics of business processes to be integrated and enables realization of business goals of underlying processes
- Focuses on Mediation Module construction

Fundamental Enterprise Service Bus Pattern

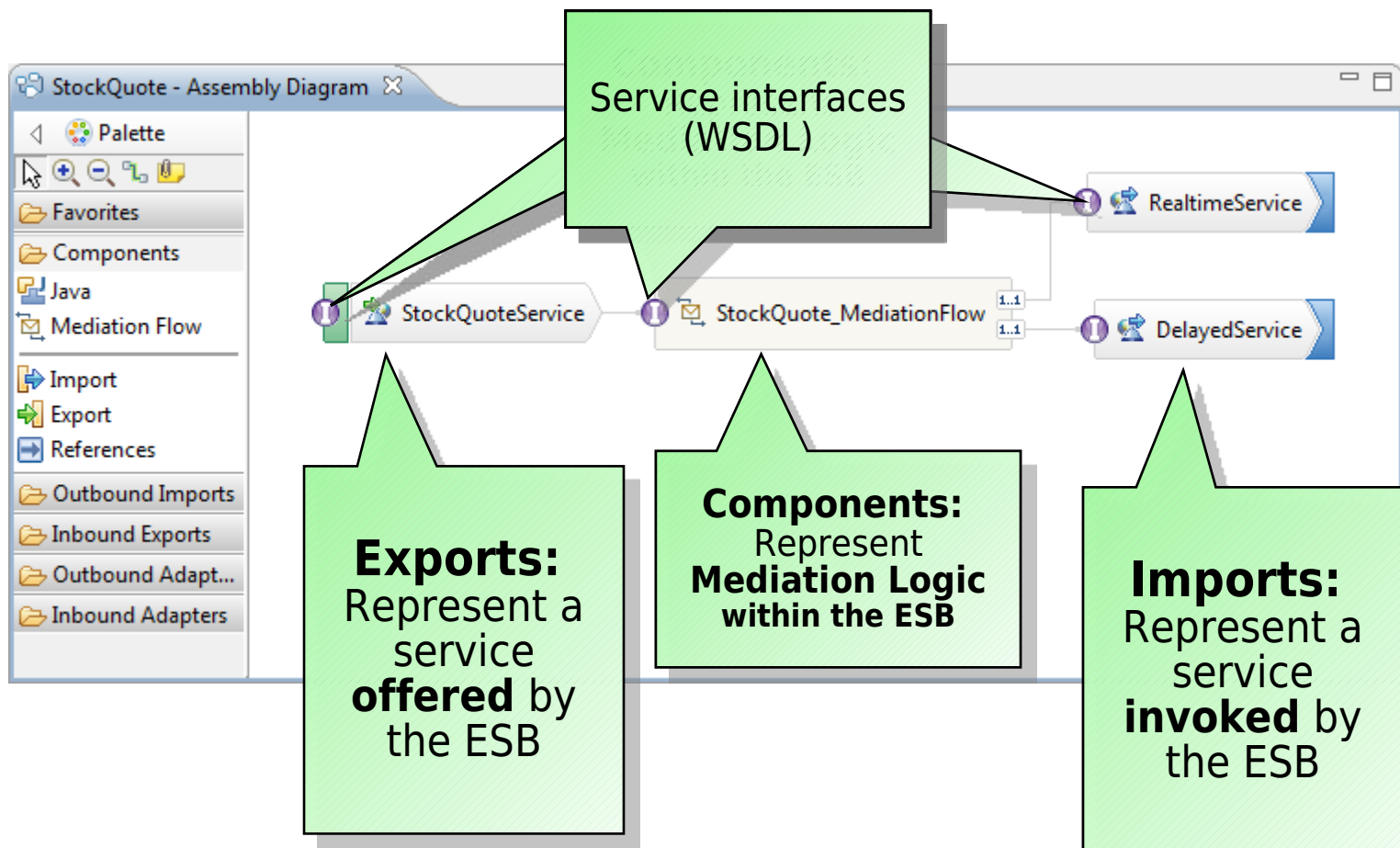


Top Level view: Imports, Exports and Components

Components are where the main connectivity logic happens

Imports and **exports** support different protocols

All interactions are represented using **service interfaces (WSDL)**



Connectivity Support

Defined as **Export** and **Import** nodes in application modules.

Supported Connectivity Bindings:

- Native **SCA** (module to module)
- **HTTP** Access (REST, XML/HTTP)
- **Web Services** (SOAP/HTTP, SOAP/JMS)
- Messaging (**JMS**, **MQ**, JMS MQ, generic JMS 3rd party adapters)
- JEE Applications (Session **EJB**)
- Java Connectivity Architecture (**JCA**) adapters

Built-in

Supplied Adapters:

- IBM WebSphere Adapter for **Email**
- IBM WebSphere Adapter for **FTP**
- IBM WebSphere Adapter for **Flat Files**
- IBM WebSphere Adapter for **JDBC**
- IBM WebSphere Adapter for **System i**
- IBM **CICS** ECI Resource Adapter
- IBM **IMS** Connector for Java

-
- IBM WebSphere Adapter for **JD Edwards** EnterpriseOne®
 - IBM WebSphere Adapter for **Oracle**® E-Business Suite
 - IBM WebSphere Adapter for **PeopleSoft**
 - IBM WebSphere Adapter for **SAP**® Software
 - IBM WebSphere Adapter for **Siebel**® Business Applications
 - IBM WebSphere Adapter for Lotus **Domino**

Converting (between transport protocols)



Business Integration - ProtocolConversion - Assembly Diagram - IBM WebSphere Integration Developer 6.1 - C:\Stephen\work\workspaces\WID61GA_...

File Edit Navigate Search Project Data Run Window Help

Business Integr... Physical Resour... ProtocolConversion - Assembly Diagram

Palette

- Favorites
- Components
- Mediation Flow
- Java
- Import
- Export
- References
- Outbound Adapters
- Inbound Adapters

Export1 Import1

The business integration perspective is used for developing WebSphere ESB modules

References Outline Visual Snippets

Build Activities Properties Problems Servers

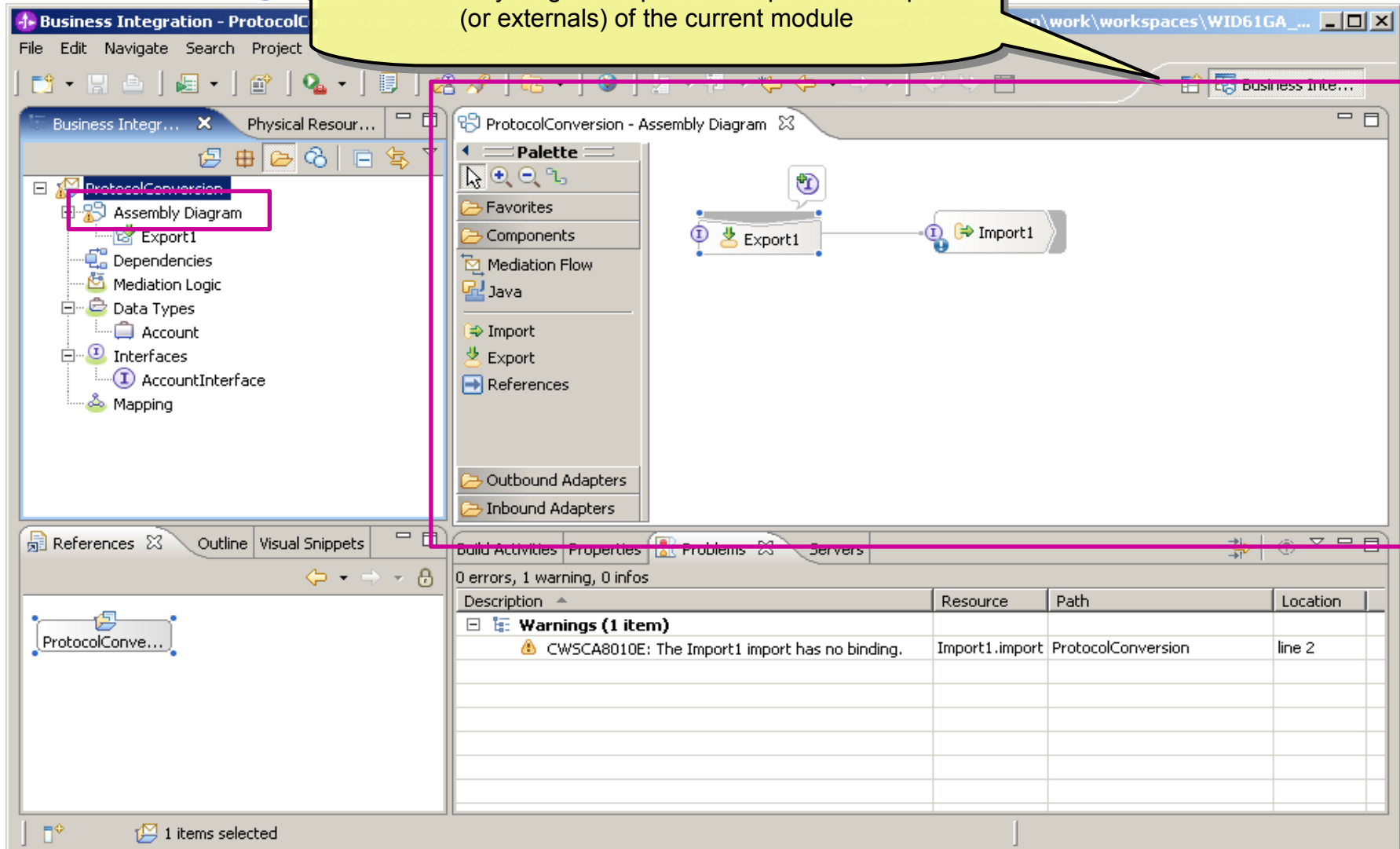
0 errors, 1 warning, 0 infos

Description	Resource	Path	Location
Warnings (1 item)			
CWSCA8010E: The Import1 import has no binding.	Import1.import	ProtocolConversion	line 2

1 items selected

Converting

The assembly diagram depicts the imports and exports (or externals) of the current module



The screenshot shows the IBM Integration Developer environment. The main workspace displays an Assembly Diagram for the 'ProtocolConversion' module, featuring an 'Export1' component connected to an 'Import1' component. The 'Palette' on the left lists various components like 'Import', 'Export', and 'References'. The 'Project Explorer' on the left shows the project structure, with 'Assembly Diagram' selected. The 'Problems' view at the bottom right displays a warning: 'CWSCA8010E: The Import1 import has no binding.' The table below details this warning.

Description	Resource	Path	Location
Warnings (1 item)			
⚠ CWSCA8010E: The Import1 import has no binding.	Import1.import	ProtocolConversion	line 2

Converting (between transport protocols)



Here we can see that one import and one export have been dragged from the palette onto the canvas and linked together

The screenshot shows the IBM Integration Developer interface. On the left, a project tree for 'ProtocolConversion' includes 'Export1' and 'Import1'. The central canvas shows 'Export1' and 'Import1' components connected by a line. A palette on the right contains 'Import' and 'Export' options. At the bottom, a 'Warnings' table displays a warning for 'Import1 import has no binding'.

Description	Resource	Path	Location
Warnings (1 item)			
CWSCA8010E: The Import1 import has no binding.	Import1.import	ProtocolConversion	line 2

A warning appears to inform us that the import has no binding

Converting (between transport protocols)



Adding a binding is a simple right click and menu selection

The screenshot shows the IBM Business Integration Studio interface. On the left, a project tree shows 'ProtocolConversion' with sub-items like 'Assembly Diagram', 'Export1', 'Dependencies', 'Mediation Logic', 'Data Types', 'Account', 'Interfaces', 'AccountInterface', and 'Mapping'. The 'Export1' component is highlighted in the main workspace. A right-click context menu is open over 'Export1', with 'Generate Binding...' selected. This has opened a sub-menu with options: 'HTTP Binding', 'Messaging Binding...', 'SCA Binding', and 'Web Service Binding'. The 'Web Service Binding' option is highlighted. The bottom of the screen shows a 'Problems' view with a warning: 'CWSCA8010E: The Import1 import...'. A table below the warning shows the location of the error.

Path	Location
protocolConversion	line 2

Converting (between transport protocols)



The screenshot displays the IBM Business Integration Designer interface. The main workspace shows an assembly diagram with two components: 'Export1' and 'Jms Import1', connected by a line. A yellow callout bubble points to this diagram with the text: 'The export has a Web service binding. The import has a JMS binding. Protocol conversion complete!'. Below the workspace, the 'Properties' view is open for 'Export1 (Web Service Binding)'. It shows fields for 'Address', 'Port', 'Service', and 'Namespace', each with a corresponding 'Configure...' or 'Browse...' button. A second yellow callout bubble points to the properties view with the text: 'Binding properties are easy to configure. The relevant options can be overridden at runtime via the admin console'. The interface also shows a project tree on the left with 'ProtocolConversion' containing 'Assembly Diagram', 'Export1', 'Jms Import1', and 'Dependencies'. The bottom toolbar includes 'References', 'Outline', 'Visual Snippets', 'Build Activities', 'Properties', 'Problems', and 'Servers'.

The export has a Web service binding. The import has a JMS binding. Protocol conversion complete!

Binding properties are easy to configure. The relevant options can be overridden at runtime via the admin console

Export: Export1 (Web Service Binding)	
Address:	<input type="text" value="http://localhost:9080/ProtocolConversionWeb/sca/Export1"/> <input type="button" value="Configure..."/>
Port:	<input type="text" value="Export1_AccountInterfaceHttpPort"/> <input type="button" value="Browse..."/>
Service:	<input type="text" value="Export1_AccountInterfaceHttpService"/>
Namespace:	<input type="text" value="http://ProtocolConversion/AccountInterface/Binding"/>

Matching and Routing: The high level view

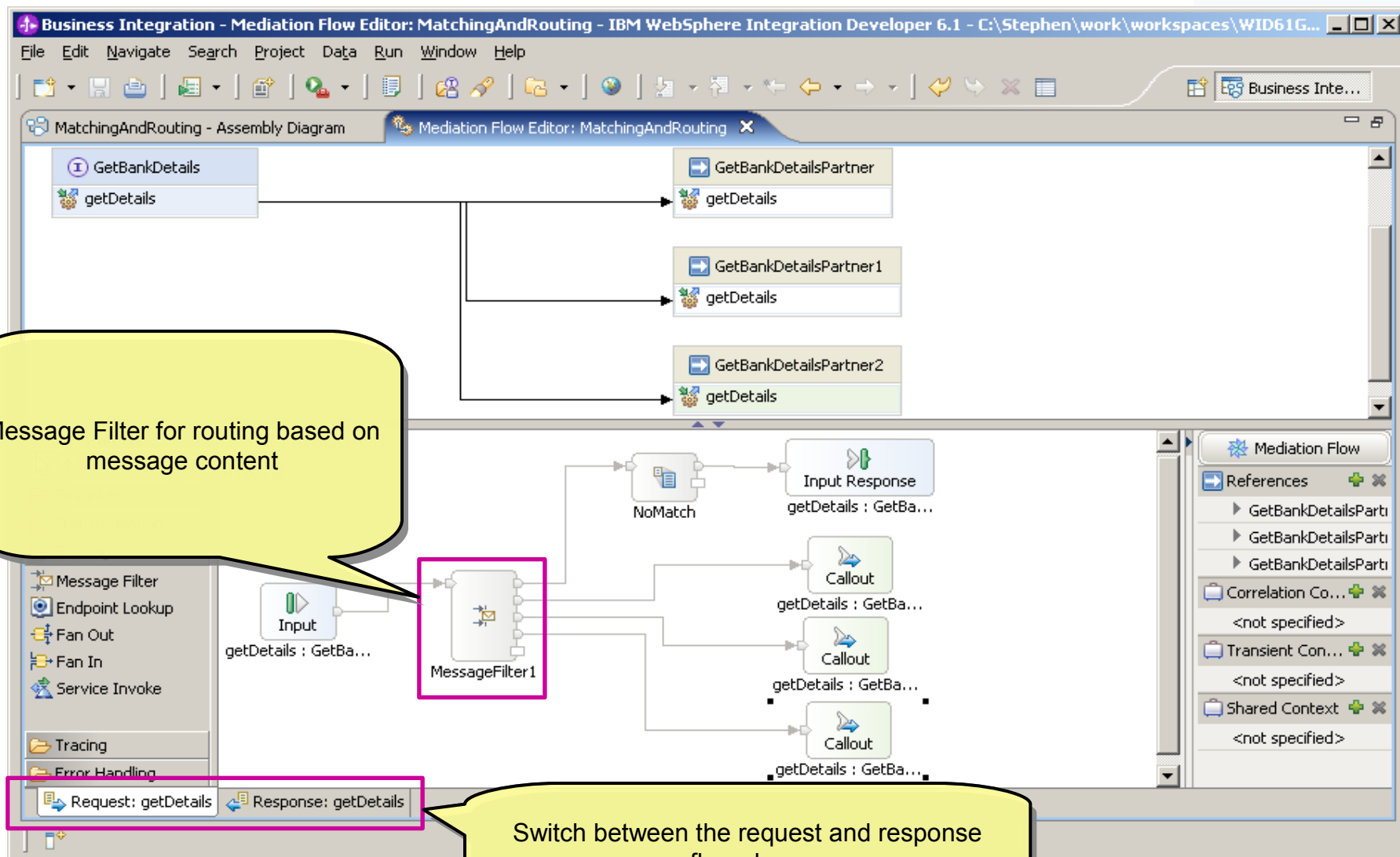


The assembly diagram shows one web service export, one mediation flow, and three imports. The mediation flow will route to the one of the three services based on the content of the input message.

The screenshot displays the IBM Integration Developer interface. The main workspace shows an assembly diagram with a central 'MatchingAndRouting' mediation flow component. This component is connected to three service import components labeled 'Bank1', 'Bank2', and 'Bank3'. To the left, a 'GetBankDetailsExport1' web service export component is connected to the 'MatchingAndRouting' component. The diagram is highlighted with a pink border. The left-hand side shows a project tree with 'MatchingAndRouting' selected. The bottom pane shows a 'Warnings' table with the following content:

Description	Resource	Path	L
Warnings (7 items)			
CW5CA8010E: The Bank1 import has no binding.	Bank1.import	MatchingAndRouting	lir
CW5CA8010E: The Bank2 import has no binding.	Bank2.import	MatchingAndRouting	lir
CW5CA8010E: The Bank3 import has no binding.	Bank3.import	MatchingAndRouting	lir

Matching and Routing: Inside the mediation component



Message Filter for routing based on message content

Switch between the request and response flows here

Matching and Routing

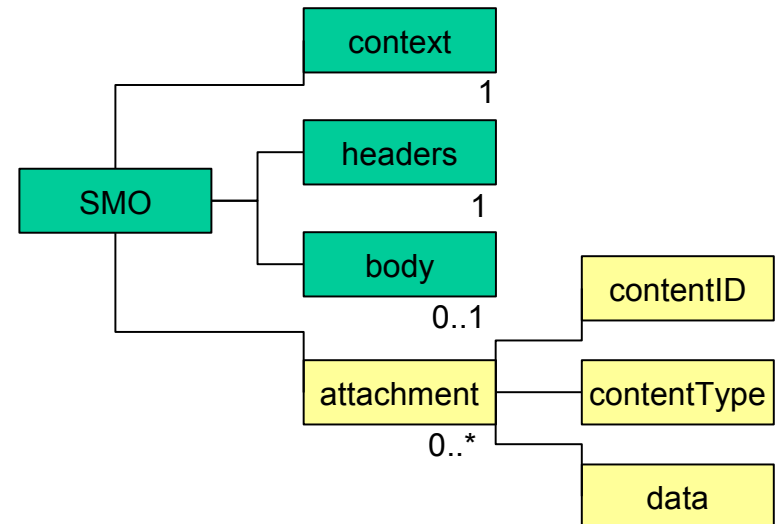


Filtering is based on XPath expressions. In business terms we are taking data from the input request and selecting a service based on that data.

Pattern	Terminal name
/body/getDetails/input/SortCode='998877'	bank1
/body/getDetails/input/SortCode='332211'	bank3
/body/getDetails/input/SortCode='445566'	bank2

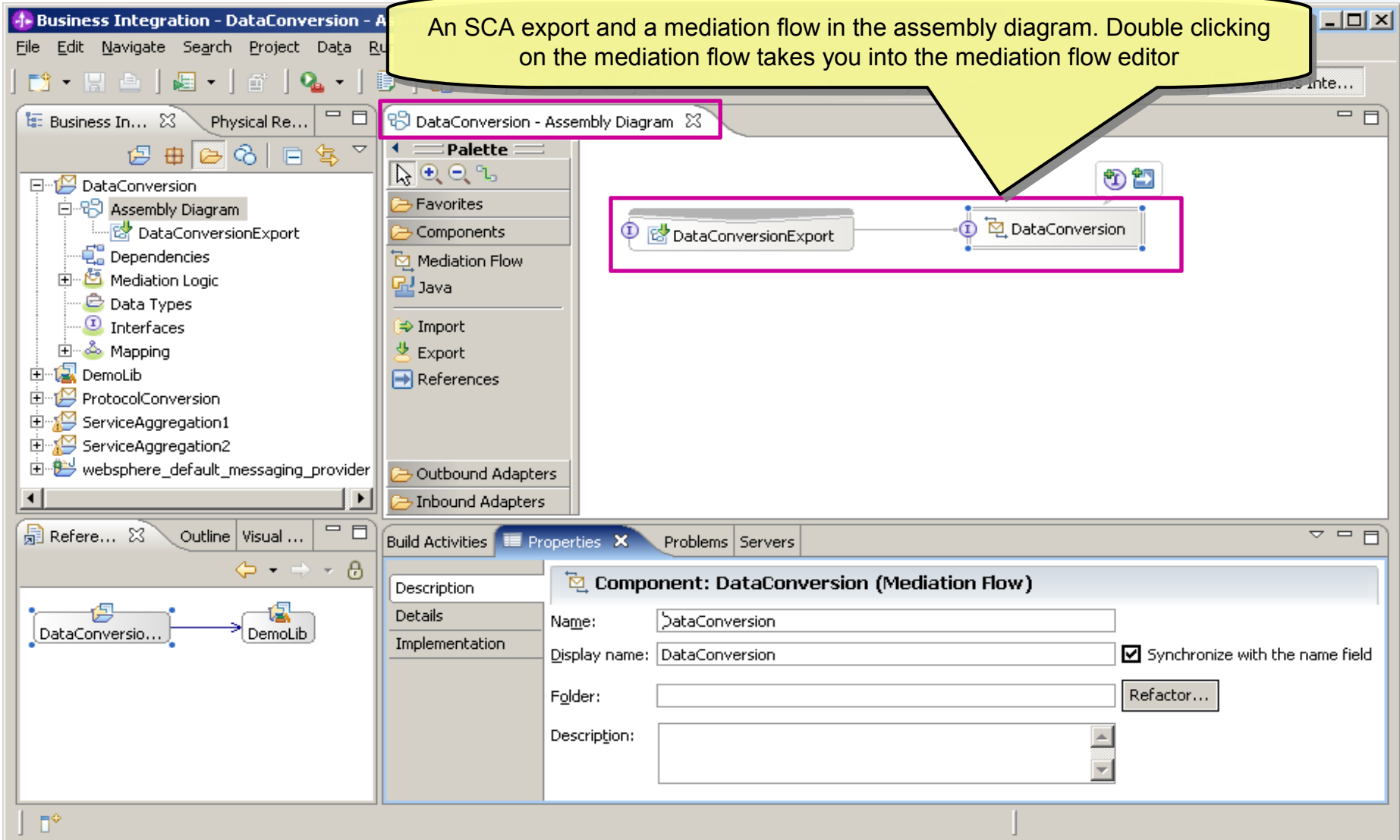
The Service Message Object

- The Service Message Object (SMO) is a 'one stop shop' for all message and context data
- It provides an abstraction of the concrete message, and can be manipulated using XPath and XSLT
 - For **all message formats** – not just XML
- Support for binary attachments



Data Conversion

An SCA export and a mediation flow in the assembly diagram. Double clicking on the mediation flow takes you into the mediation flow editor



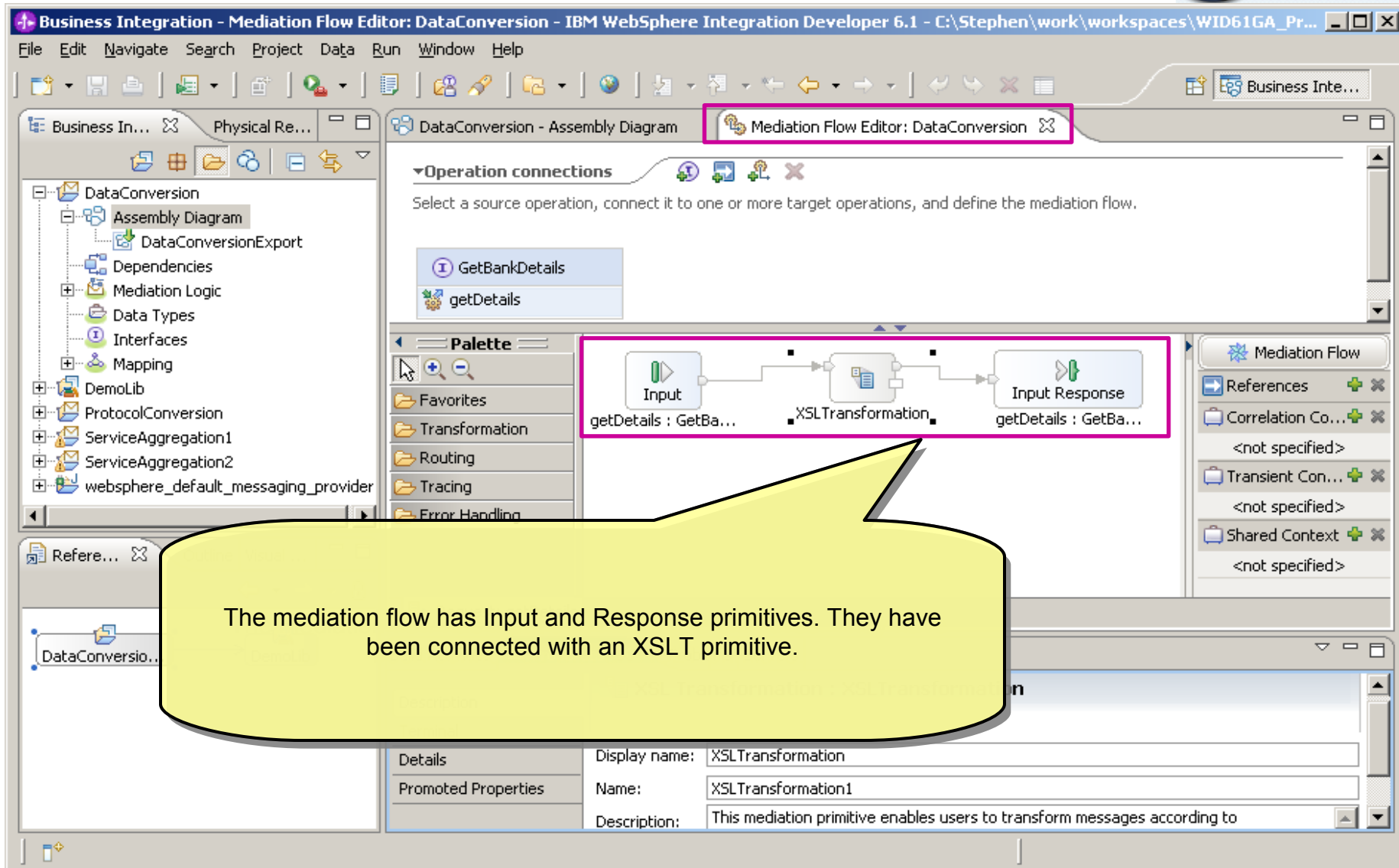
The screenshot displays the IBM Integration Developer interface. The top window, titled "DataConversion - Assembly Diagram", shows a diagram with two components: "DataConversionExport" and "DataConversion", connected by a line. A yellow callout bubble points to this diagram with the text: "An SCA export and a mediation flow in the assembly diagram. Double clicking on the mediation flow takes you into the mediation flow editor".

The bottom window, titled "Properties", shows the configuration for the selected component: "Component: DataConversion (Mediation Flow)". The fields are as follows:

Field	Value
Name:	DataConversion
Display name:	DataConversion <input checked="" type="checkbox"/> Synchronize with the name field
Folder:	<input type="text"/> Refactor...
Description:	<input type="text"/>

The left sidebar shows a project tree with "DataConversion" expanded to "Assembly Diagram". The bottom-left window shows a smaller version of the assembly diagram with "DataConversion" and "DemoLib" components.

Data Conversion



The screenshot shows the Mediation Flow Editor interface. The main workspace displays a mediation flow with three primitives: 'Input' (labeled 'getDetails : GetBa...'), 'XSLTransformation', and 'Input Response' (labeled 'getDetails : GetBa...'). These are connected in a sequence. A yellow callout bubble points to this flow with the text: 'The mediation flow has Input and Response primitives. They have been connected with an XSLT primitive.'

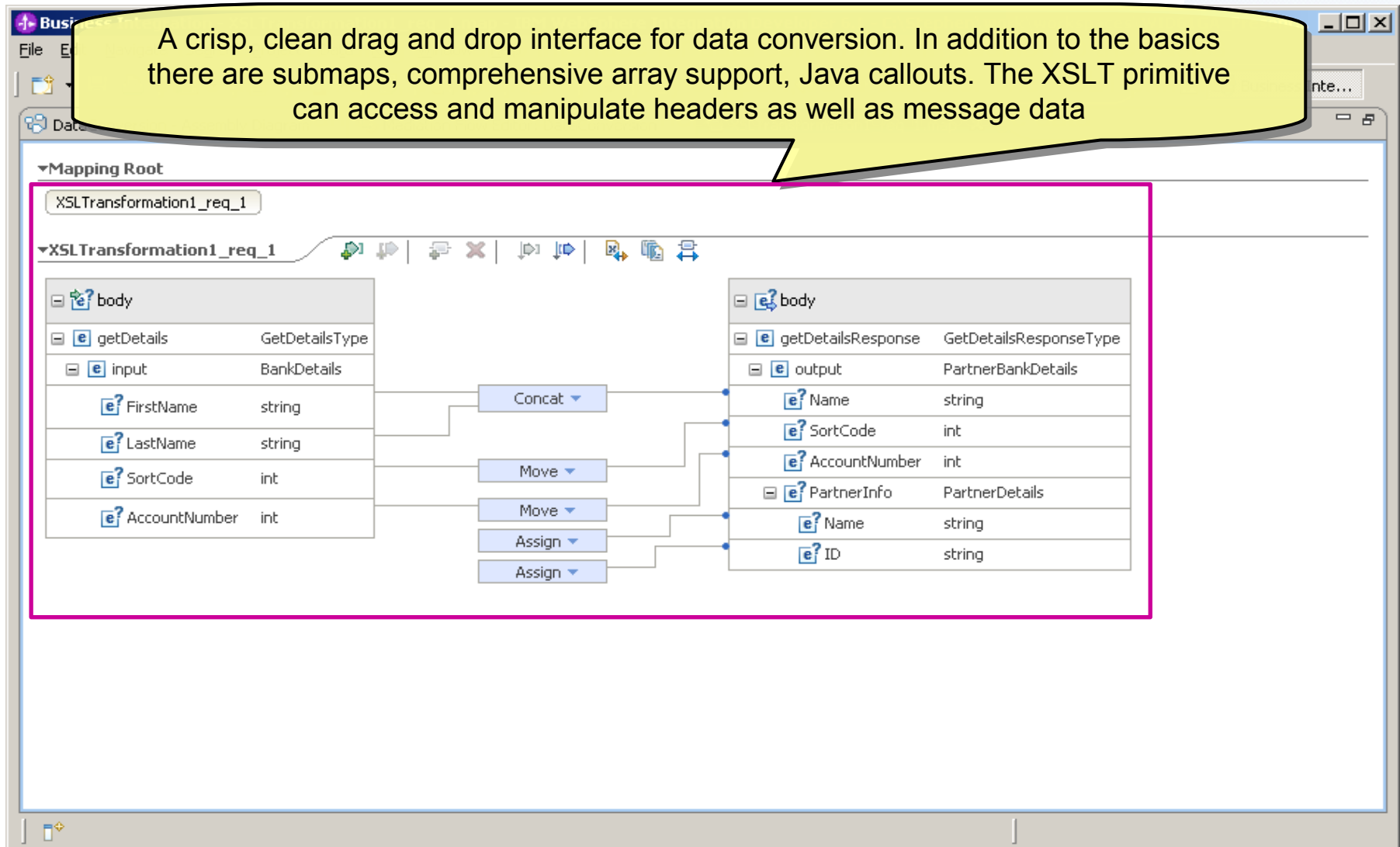
The interface includes a menu bar (File, Edit, Navigate, Search, Project, Data, Run, Window, Help), a toolbar, and several panels:

- Left Panel:** Project Explorer showing a tree view of 'DataConversion' with sub-items like Assembly Diagram, DataConversionExport, Dependencies, Mediation Logic, Data Types, Interfaces, Mapping, DemoLib, ProtocolConversion, ServiceAggregation1, ServiceAggregation2, and websphere_default_messaging_provider.
- Top Center Panel:** 'Operation connections' section with a list containing 'GetBankDetails' and 'getDetails'.
- Bottom Left Panel:** 'Palette' with categories: Favorites, Transformation, Routing, Tracing, and Error Handling.
- Right Panel:** 'Mediation Flow' configuration area with sections for References, Correlation Co..., Transient Con..., and Shared Context, each with expand/collapse icons.
- Bottom Panel:** Properties view for the selected 'XSLTransformation' primitive, showing:

Details	Display name:	XSLTransformation
Promoted Properties	Name:	XSLTransformation1
	Description:	This mediation primitive enables users to transform messages according to

Data Conversion

A crisp, clean drag and drop interface for data conversion. In addition to the basics there are submaps, comprehensive array support, Java callouts. The XSLT primitive can access and manipulate headers as well as message data

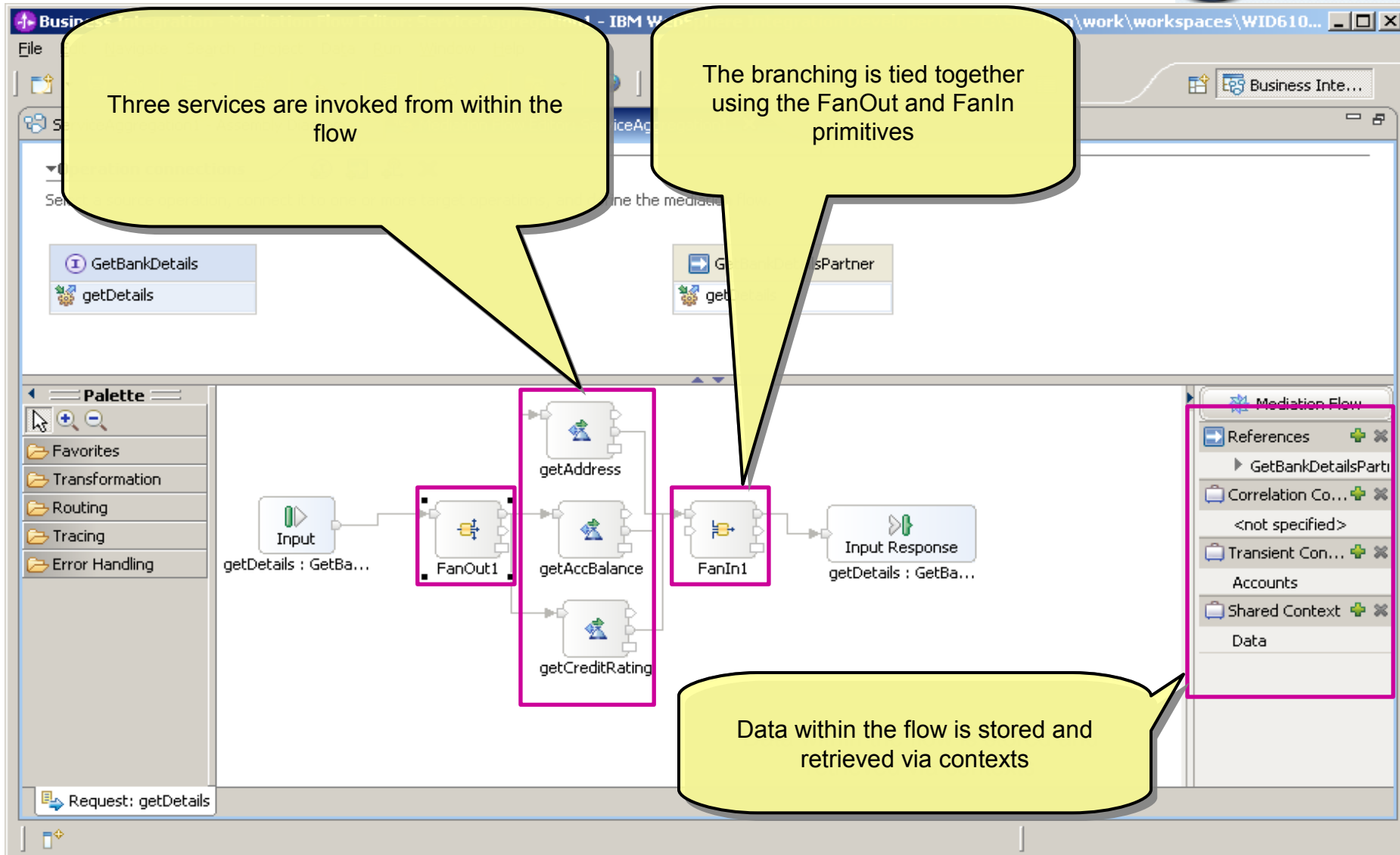




Service Aggregation

- Split messages for separate processing
- Augment messages from multiple services
- Amalgamate data from multiple sources
- Provide alternate services for failover

Service Aggregation

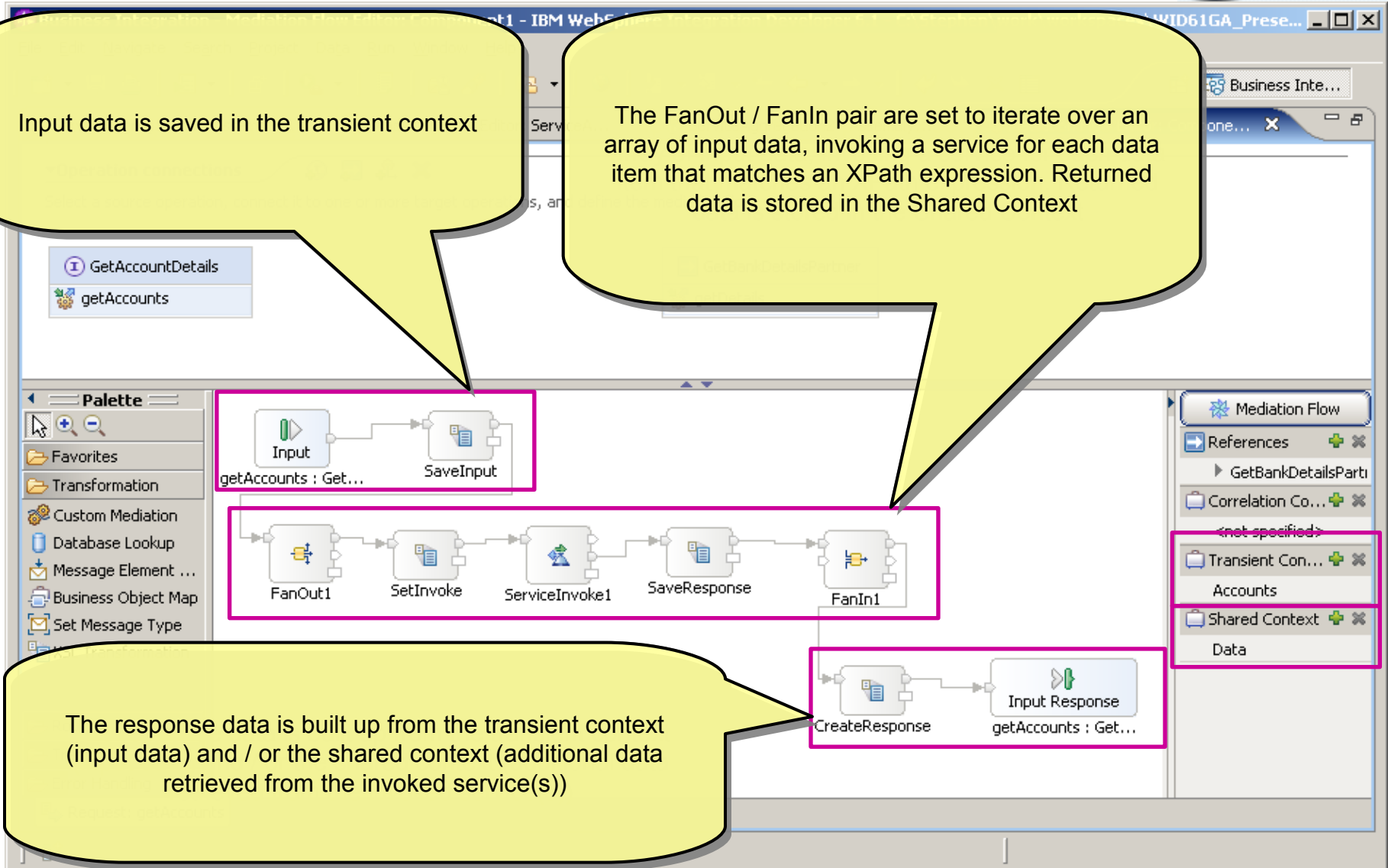


Service Aggregation



Input data is saved in the transient context

The FanOut / FanIn pair are set to iterate over an array of input data, invoking a service for each data item that matches an XPath expression. Returned data is stored in the Shared Context



The response data is built up from the transient context (input data) and / or the shared context (additional data retrieved from the invoked service(s))

WSRR & Dynamic Endpoint Lookup: Top Level View



Main task of the mediation component is to perform dynamic routing

Scenario: Routing insurance claims to named insurance providers

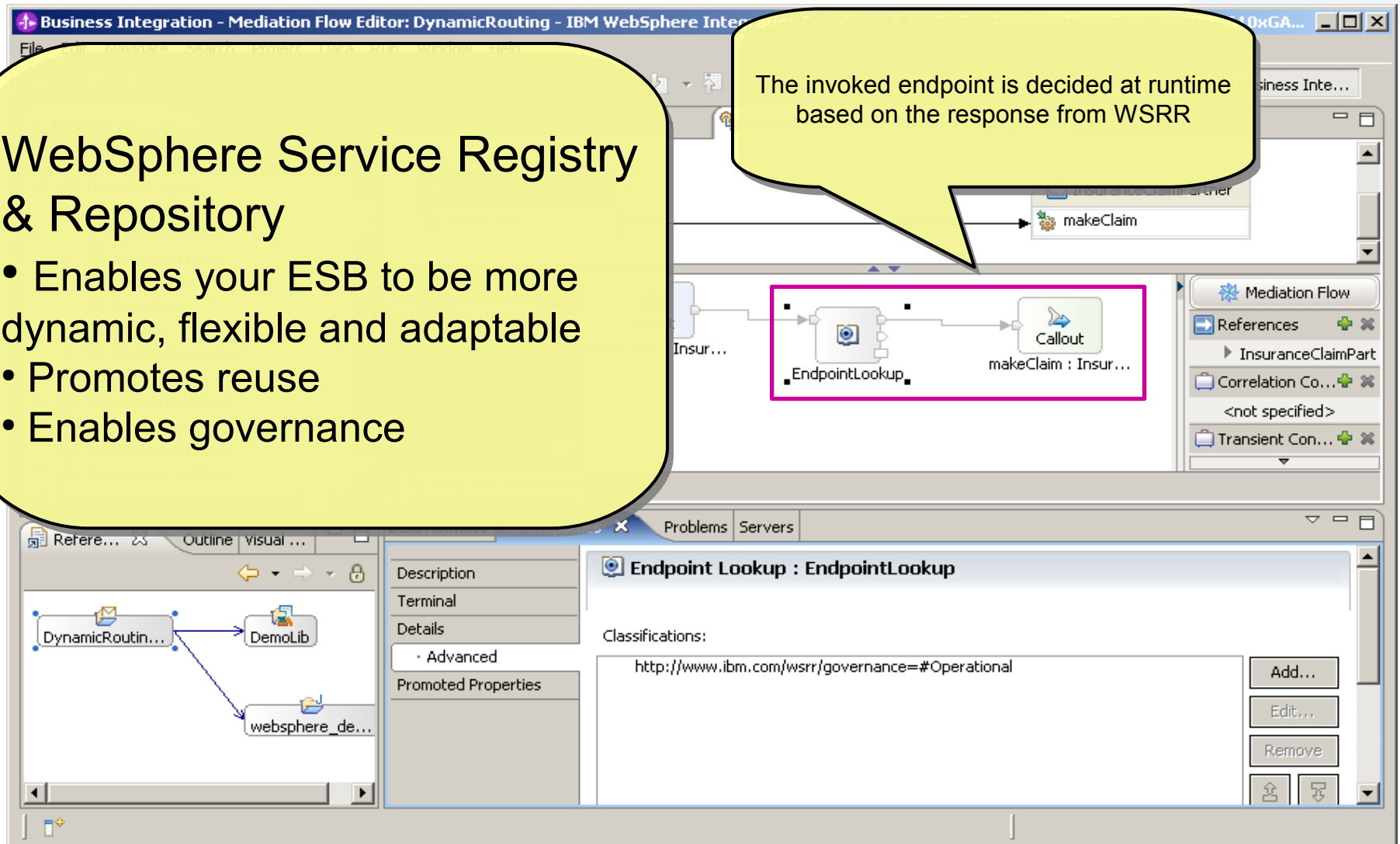
- Insurance providers support a common web service
- Only difference between providers is the endpoint address
- Providers may be added, deleted or changed at any time
- Unsuccessful WSRR queries are forwarded to a JMS queue for resolution

WSRR & Dynamic Endpoint Lookup: Inside the mediation

WebSphere Service Registry & Repository

- Enables your ESB to be more dynamic, flexible and adaptable
- Promotes reuse
- Enables governance

The invoked endpoint is decided at runtime based on the response from WSRR



The screenshot displays the IBM WebSphere Integration Developer interface. The main window shows a mediation flow editor with a process flow starting from an input, passing through an **EndpointLookup** component (highlighted with a pink box), and then a **Callout** component labeled **makeClaim : Insur...**. A yellow callout bubble points to the **makeClaim** component with the text: "The invoked endpoint is decided at runtime based on the response from WSRR".

The bottom right pane shows the configuration for the **Endpoint Lookup : EndpointLookup** component. It includes a "Classifications" section with the URL: `http://www.ibm.com/wsrr/governance=#Operational`. Buttons for **Add...**, **Edit...**, and **Remove** are visible.

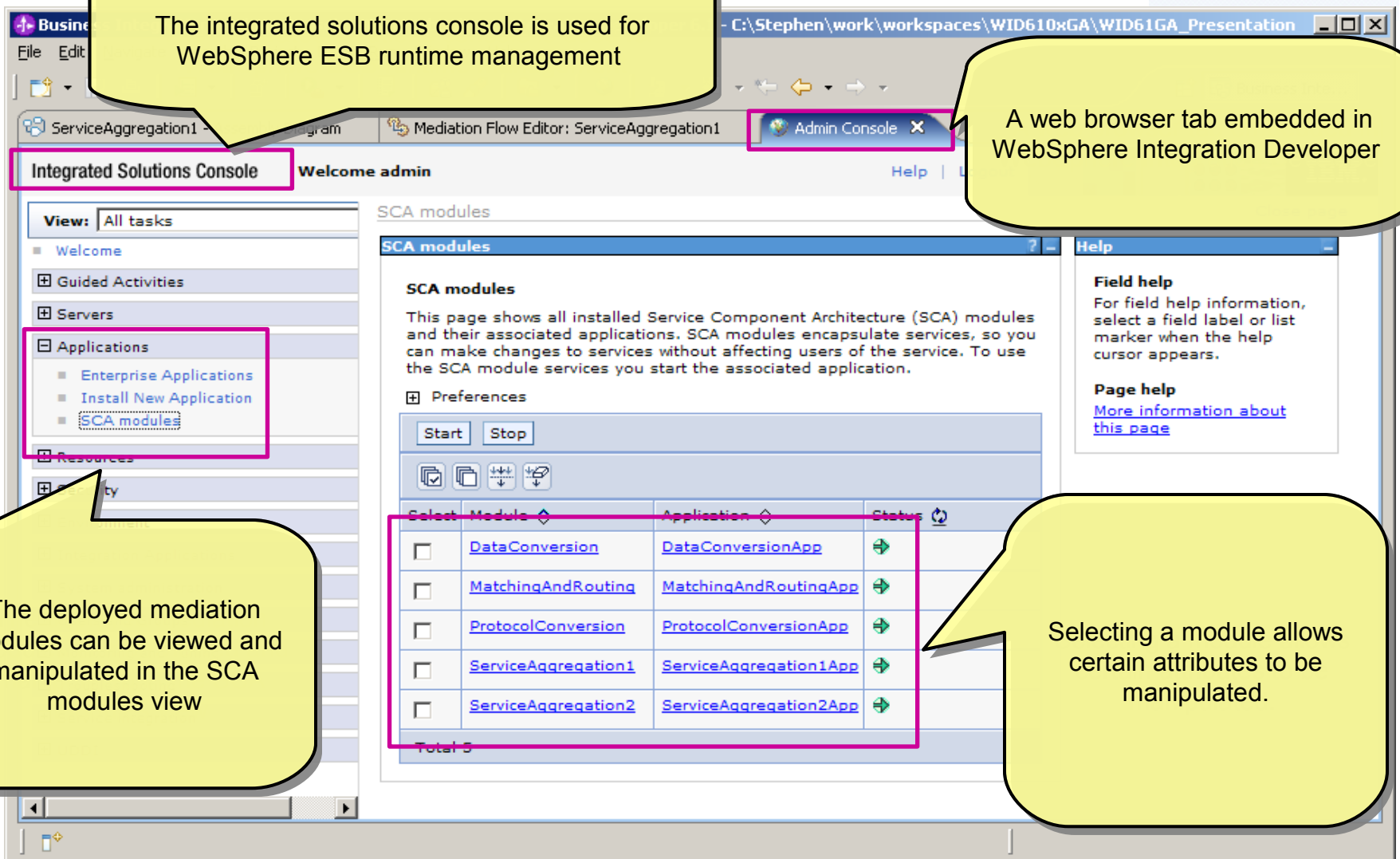
The bottom left pane shows a project outline with components like **DynamicRoutin...**, **DemoLib**, and **websphere_de...**.

Runtime Management



**Solution
Administrator**





The integrated solutions console is used for WebSphere ESB runtime management

A web browser tab embedded in WebSphere Integration Developer

The deployed mediation modules can be viewed and manipulated in the SCA modules view

Selecting a module allows certain attributes to be manipulated.

Integrated Solutions Console Welcome admin

View: All tasks

- Welcome
- Guided Activities
- Servers
- Applications
 - Enterprise Applications
 - Install New Application
 - SCA modules
- Resources
- ty

SCA modules

SCA modules

This page shows all installed Service Component Architecture (SCA) modules and their associated applications. SCA modules encapsulate services, so you can make changes to services without affecting users of the service. To use the SCA module services you start the associated application.

Preferences

Start Stop

Select	Module	Application	Status
<input type="checkbox"/>	DataConversion	DataConversionApp	➔
<input type="checkbox"/>	MatchingAndRouting	MatchingAndRoutingApp	➔
<input type="checkbox"/>	ProtocolConversion	ProtocolConversionApp	➔
<input type="checkbox"/>	ServiceAggregation1	ServiceAggregation1App	➔
<input type="checkbox"/>	ServiceAggregation2	ServiceAggregation2App	➔

Total 5

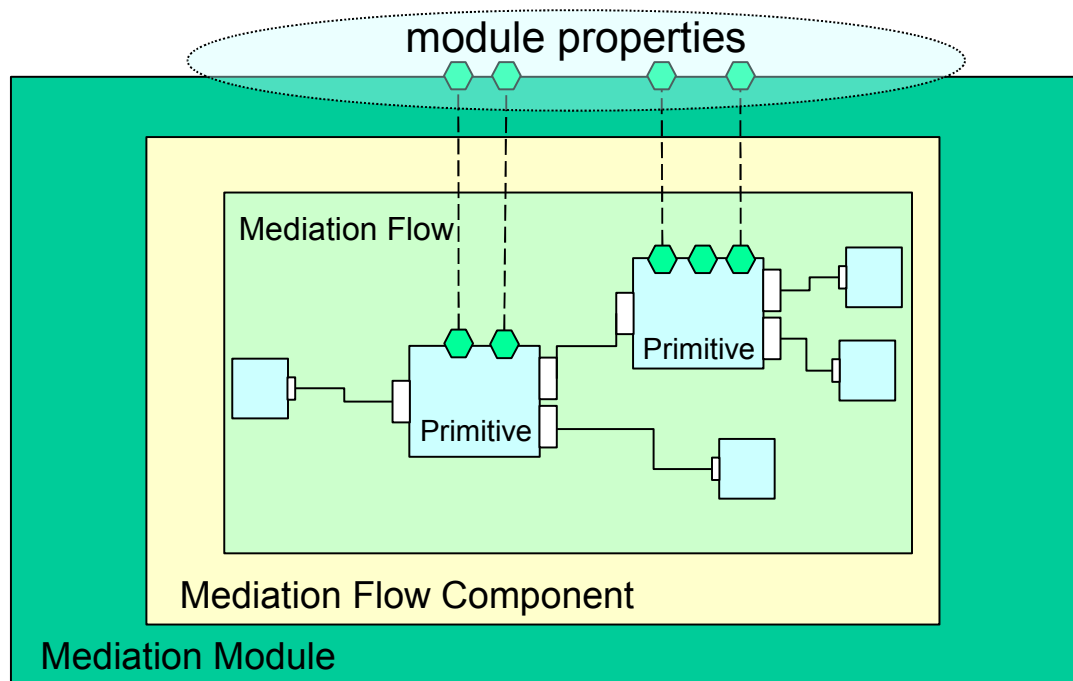
Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
[More information about this page](#)

Mediation Control: Promoted Properties



- The Mediation Flow developer can choose to **promote** the property of a mediation primitive, and can select a new name (alias) for this promoted property
- When this is done, the promoted property becomes available for the administrator to override as **Module Properties**
- Can be placed under the control of WSRR as **Mediation Policies**
- This enables to integration developer to **selectively** empower the administrator to be able to customize some aspects of mediation behaviour

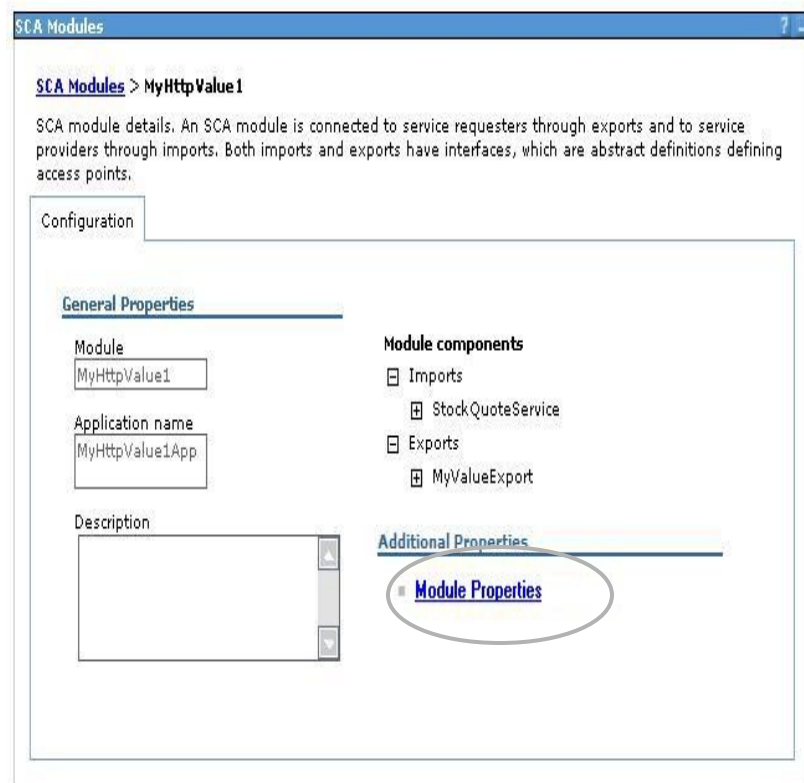


ESB Mediation Module Properties



Property	Promoted	Alias	Alias value
Transaction mode	<input type="checkbox"/>		
Root	<input checked="" type="checkbox"/>	MessageLogger.1.root	/body

SCA Modules



SCA Modules > MyHttpValue1

SCA module details. An SCA module is connected to service requesters through exports and to service providers through imports. Both imports and exports have interfaces, which are abstract definitions defining access points.

Configuration

General Properties

Module: MyHttpValue1

Application name: MyHttpValue1App

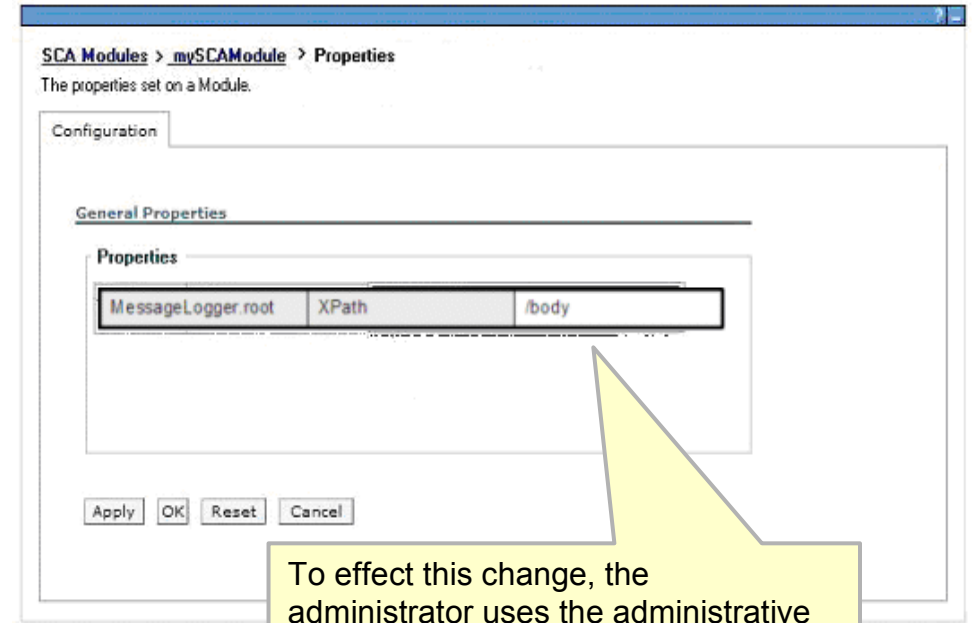
Description:

Module components

- Imports
 - StockQuoteService
- Exports
 - MyValueExport

Additional Properties

- Module Properties



SCA Modules > mySCAModule > Properties

The properties set on a Module.

Configuration

General Properties

Properties

MessageLogger.root	XPath	/body
--------------------	-------	-------

Apply OK Reset Cancel


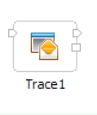
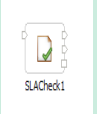
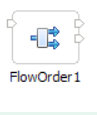
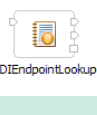
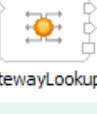
To effect this change, the administrator uses the administrative console to view the module properties defined for the mediation module, and modifies the value of the MessageLogger.root module property from /body to /

New in Version 7.0

- 7.0: Base Connectivity Enhancements
 - EJB export, SOAP with Attachments

- 7.0: Enhanced Mediation Support
 - Additional Pattern Support
 - Federated service connectivity enablement
 - Proxy gateway
 - Mediation subflow, policy enhancements
 - New QoS: Event sequencing, Store and Forward
 - New Mediation primitives

WebSphere ESB V7.0: New Primitives

	Primitive	Description			
	Message Validator	Validate the message matches the schema			
	Trace	Print a trace message to the server log			
	SLA Check	Check whether the consumer of this target service has an agreement to use it defined in WSRR.			
	Flow Order	Control the order of execution for multiple flow branches			
	UDDI Endpoint Lookup	Lookup service endpoint in UDDI			
	Gateway Endpoint Lookup	Lookup service endpoint based on a virtual service name defined in WESB			

WebSphere ESB V7.0 Mediation Policy Enhancements

Mediation Policy Administration

Policy attachment: MyFirst WSRR definition used: blade31
 Endpoint: bs_http_BasicServiceHttpPort

Group Name Selection

Group Name	Supported Modules
DomainD	Mediation5 Mediation4 Mediation1 Mediation2
DomainA	Mediation3 Mediation1 Mediation2
DomainB	Mediation4 Mediation1
DomainC	Mediation3

Policy Selection

Use existing

Select a policy

- Mediation1-DomainD.xml#DomainD_DefaultPolicy
- Mediation4-DomainD.xml#DomainD_DefaultPolicy
- uri:brian_pol_attach4_DomainD
- uri:brian_pol_attach2_DomainD
- uri:brian_pol_attach1_DomainD
- Mediation2-DomainD.xml#DomainD_DefaultPolicy
- uri:brian_pol_attach10_DomainD
- Mediation5-DomainD.xml#DomainD_DefaultPolicy

Policy Administration

Attachment: MyFirst
 Policy: MyNewPolicyD
 Endpoint: bs_http_BasicServiceHttpPort

Mediation Policy Administration

Policy attachment: MyFirst WSRR definition used: blade31
 Policy: MyNewPolicyD
 Endpoint: bs_http_BasicServiceHttpPort

Assertions

Group Name	Property Name	Value
DomainD	Dalpha	true

Group name: DomainD Property name: Value:

Gate Conditions (Optional)

Name	Value
medGate_COND1	CustomerID = IBM

Gate condition name: medGate_ Value:

Name	Property Name	Value
DomainD	Dalpha	true

Group name: DomainD Property name: Value:

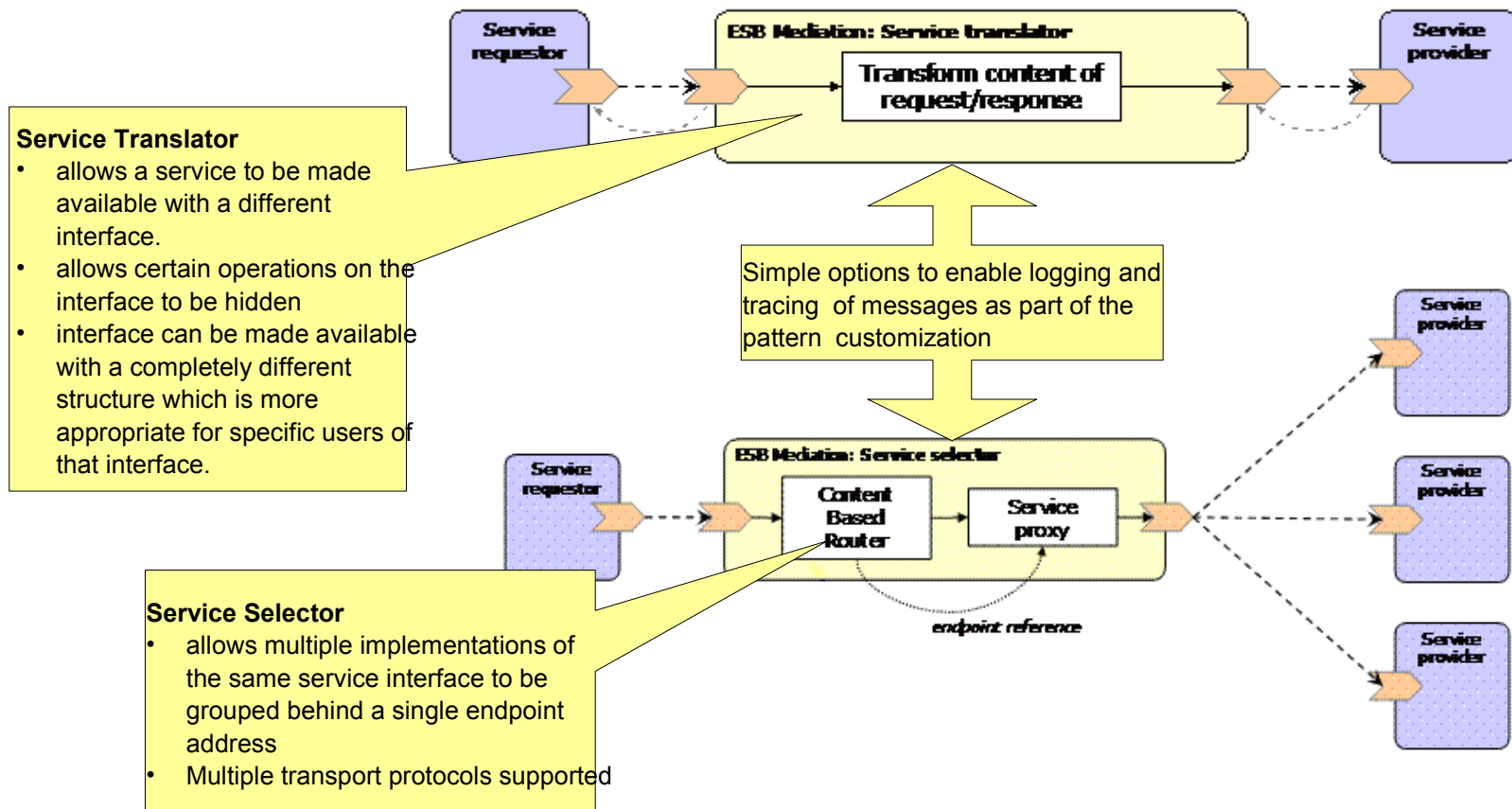
Gate Conditions (Optional)

Name	Value
medGate_COND1	CustomerID = IBM

Gate condition name: medGate_ Value:

Featurepack Updates: New Mediation Pattern Support

- Featurepack:
 - Availability Date: June 11th 2010
- Accelerators for two more common integration patterns
 - Creates complete modules, ready for deployment to WESB or WPS
 - Capability Delivered in WebSphere Integration Developer, supported by WebSphere ESB and WebSphere Process Server



WebSphere Enterprise Service Bus Roadmap

IBM's plans, directions, and intent are subject to change or withdrawal

- Smart connectivity for Java applications and BPM
- Based on WAS v7 (JDK 6, Java EE 5, EJB 3.0), leveraging SCA FeP, XML FeP, Eclipse 3.4, MQ v7
- WS enhancements: WS-Addressing, WS-Policy, Kerberos token profile, SOAP Attachments
- Store-and-forward and Health monitoring widgets
- Web-based widgets for mediation policies
- Supports WSRR v7 including Service Federation
- Supports Adapters v7, enhanced for JDBC, FTP, Email, SAP, Oracle EBS, CICS, EMD tooling, iSeries. New adapter support for WBE, Lotus Domino
- Pattern-based mediation authoring with WID v7: Service Gateway enhancements

WebSphere ESB Registry Edition

- WESB + WSRR Getting started package
- Dynamic Service Connectivity, Reuse
- SOA Visibility: Governance, Management

**WESBRE
v7.0
8 Oct 2010**

Major release
 Minor release

Future release

.....>
Delivery of Next capabilities

WESB Future Capabilities

- Configurable ESB: Simplify Connectivity with Web UI, WSRR, applying patterns to make services ready for use by consumers
- Scriptable ESB: Textual mediation flow format for custom patterns, custom mediations, compare/merge, copy/paste, ...
- Enhance and extend common transformation and mapping

**FeP2 WID,
Adapters
Oct 2010**

- Adapter enhancements (Oracle, ECM)
- WLE calls web services

**FeP1 WID,
Adapters
June 2010**

V7.0.0.1

- Service Translator pattern
- Service Selector pattern
- Alias / Vocabulary in XML mapper
- Adapter enhancements (JDBC, iSeries, LotusDomino, Siebel)

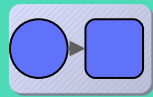
**WESB
WID, Adapters**

**V7.0
11 Dec 2009**

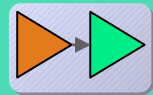
- WebSphere ESB is central to the SOA evolution
- Integrates seamlessly with WebSphere platform
- Delivers business-critical qualities of service
- Integrated solution for service mediation and hosting



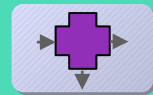
Delivers leadership in SOA standards for service composition, and leverages the embedded messaging and web services engines from WebSphere



Integrates everything with WebSphere Adapters for enterprise applications, the breadth of the WebSphere ecosystem, and support for standard protocols



Optimized for standard XML and web services formats, with basic support for other common formats



Provides business visibility with embedded event engine for Business Activity Monitoring solutions

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Copyright and Trademarks

© IBM Corporation 2011. All Rights Reserved.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.